D0 Data Requests Test Page

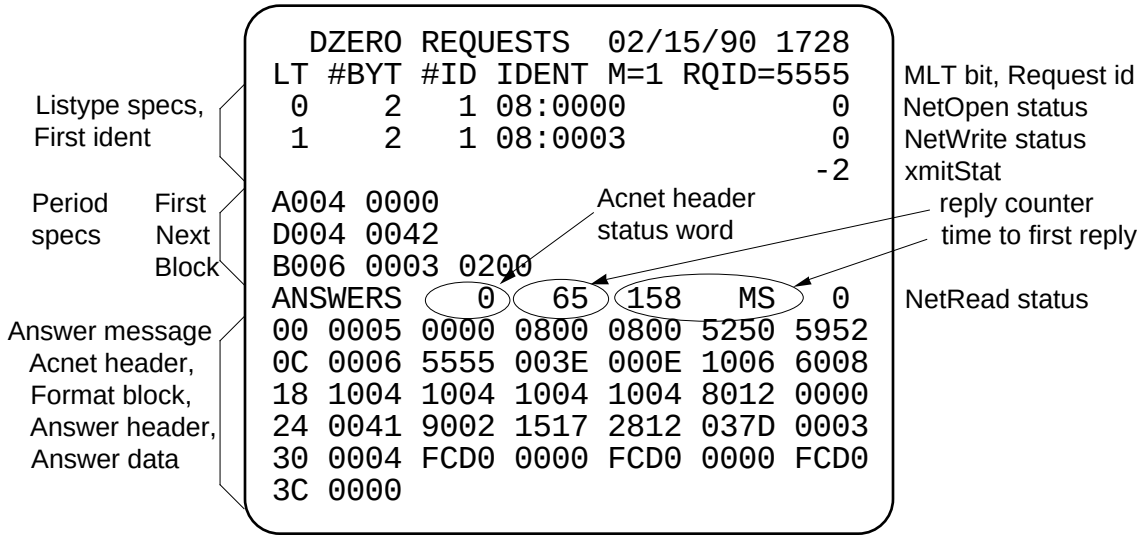
Local Station Application

Mar 21, 1990

Introduction

A new protocol for data request network messages was designed for use with the D0 detector's CDAQ (Controls Data Acquisition) software. The VME local stations support this message format for data requests. This application page exercises the new protocol by issuing data requests to a local station and displaying the responses that are returned. It is written using the local station's Network Layer interface routines.

Display page layout



This snapshot of the test program display serves as an example of several of the features it provides. It shows an example of requesting a 2-byte reading from channel 0 in node 08 and a 2-byte setting from channel 3. The period specification indicates that the data is to be collected the first time right away and at intervals of 15 Hz (66 msec) thereafter, accumulating (blocking) 3 sets of data (with a 512 msec timeout) before returning a response.

Parameter entry

The `M` bit is set for periodic requests. The request remains active until a cancel message is sent. If the `M` bit is zero, the request is automatically cancelled after update of the first set of reply data. The `RQID` is the request id that serves to uniquely identify one request from another.

Enter up to three listype specifications each with an associated starting ident. The listype#, the #bytes requested/ident, and the #idents are specified in decimal. Enter the starting ident in hex. The ident length is determined by number of

digits typed. When the #idents > 1, the additional idents are added sequentially

using a step size of 1. (For memory addresses, the step size is 2.) To omit a listype spec, blank out the listype# field.

The period specifications occupy three lines for the First, Next, and Block specs and are entered in hex. (See Al Jonckheere's document entitled "D0 Control Data Acquisition Network Data Transmission Protocol" for a complete presentation of the entire protocol including the period specification parameters.) To omit a period spec, blank out the first word of the line. The other words will remain, but the spec will not be used.

Interrupt anywhere in the parameter specification area of the screen (on rows 2–8) to initiate the data request. The word ANSWERS in the response area of the screen should be hi-lited to indicate the request is active. (If it is a one-shot request, or if an error is returned, it may not stay hi-lited for long.) When the first (or only) response is received, the elapsed time is displayed on the ANSWERS line in milliseconds. This time is measured from just *before* the call to `NetWrite` until just *after* the call to `NetRead` that returns the first response in the test program.

Just to the left of the elapsed time to first response is the count of replies received. The Acnet header status word is also shown in decimal just after the word ANSWERS. This is done for convenience in interpreting error codes, which tend to be negative numbers.

Other status replies that are shown are toward the right end of the screen. The status return from NetOpen, which is called upon entry to the page, is given at the end of the third line. Below that is the return from the call to NetWrite when the request message was issued. Below that is the transmit status word that gives the success of the network transmission. At the end of the ANSWERS line is the status return from the call to NetRead which returns the reply data.

Cancel an active request by an interrupt on the ANSWERS line. The hi-liting will be removed as the cancel message is sent (a USM with bit#9 set in the Acnet header flags word).

Answer data viewing

Six lines are used to display answer data in hex with 6 words per line. The byte value at the left shows the offset in bytes of the first word on the line. There are three ways to adjust the starting offset for the block of answers.

The easiest way is to use the raise/lower buttons on the local console. It will adjust the offset by 72 bytes (6 lines of 6 words each) at a time. If you advance too

far, it wraps to the start.

To adjust the offset so that the first word is one of the displayed data words, merely interrupt under the word you want to be the starting word displayed. Obviously, you can only move ahead in this way.

The third way to adjust the offset is by typing in the desired offset in the first characters of the first line of answer data and interrupting. Three characters can be entered, even though only the least significant two characters can be displayed due to the screen's limited number of characters per line.

The entire response message is displayed, beginning with the 9-word Acnet header. (For an error response, this is all you get.) This is followed by the format block, beginning with the size word of that block. That is followed by the 9-word answer message header. The remainder of the response message is the answer data itself.

Details of the example response

The first 9 words are the Acnet header. The first word shows that it is a reply message with the `MLT` bit set. The next word is the reply status word, and it is also shown in decimal on the line above. The destination and source bode of the request are both node 08 in this case. This example illustrates use of the network to make a request to itself. (If it didn't do that, it would always require two nodes to do the test.) A by-product of this is that the xmitStat value shown at the end of the fifth line is -2, indicating "address not recognized," which in this case is normal.

The next two words of the Acnet are the destination task name, which for D0 data requests is `RPYR`. The source task id is 6, which denotes the table index in the Network Connect Table returned by the call to `NetOpen`. The request id is followed by the message size word, which is the total size of the entire response message including the Acnet header itself.

The format block length of 14 bytes including the length word shows six format specs. The first three describe the format of the answer message header and are always present. They indicate 3 words, one 8-byte time, and 2 words. The last three format specs describe the format of the three sets of answers as given by the blocking specification. Each indicates the two words (a reading and a setting) that were requested in this example.

The answer message header starts with the `8012` word. The status word is zero. The only nonzero status here would be 4, indicating a bus error, possible when accessing arbitrary memory locations. All other errors are detected during request initialization and return error responses as status-only replies, consisting

of only the Acnet header.

The next word is the sequence counter. Note that its value matches the reply counter shown in decimal above, which is good. It is followed by the BCD time field. Note that this matches the time of day that is shown on the top line of the display. (The seventh byte of this 8-byte field is a cycle count, also in BCD, and the last byte is a residual half-millisecond count.) The time is followed by the number of data sets (=3 from the blocking specs) included in this response and the size of each set (=4 bytes in this case).

At long last we get to the answer data. Each of the three sets shows the same values of the reading and setting words. As to why the reading and setting don't match in this case, remember that node 08 is a test station, in which almost anything is possible.

The time response value for this example shows 158 msec, which may seem long. But remember that the example specified that 3 sets of 15 Hz data were to be accumulated before returning a response. The first set will be collected right away as soon as the request is processed, and the third set will be collected two cycles later. Since this station actually runs at 12.5 Hz in the absence of 15 Hz interrupt input signal, two more cycles would be 160 msec. The request was issued by the application program in the same node, and it runs soon after the data collection and update time for that node. (The 66 msec value on the period specs is rounded to one 80 msec cycle.) So that's why. A more typical simple example without blocking would result in an elapsed time about 5–6 msec for the first response. Interestingly, this matches the analogous response time that can be measured for the "classic" data request protocol used by the Local Stations.